

Listing of Claims:

1. (Currently Amended) A method for simulating application workload on an e-business application server hosting a plurality of e-business application programs, comprising:

forwarding a ~~placebo~~ simulated transaction work request to the e-business application server hosting a plurality of application programs, wherein the ~~placebo~~ simulated transaction work request includes a priority indicator and results in a load being applied to the e-business application server;

~~applying a load for said placebo transaction work request resulting from said executing step to said e-business application server;~~

receiving said forwarded ~~placebo~~ simulated transaction work request by a workload driver;

said workload driver automatically translating said forwarded ~~placebo~~ simulated transaction work request into [[a]] computer program operation— operations to be performed in the e-business application server based on the priority indicator, wherein said automatically translating step is determined by configuration information contained within a configuration file, said configuration file associating computer program commands with transaction work requests, and wherein reconfiguring said configuration file varies a processing complexity and duration of at least one of the computer program commands; and

executing said computer program operation in the e-business application server.

2. (Currently amended) The method according to claim 1, wherein said translating step further comprises:

determining a workload based on said forwarded ~~placebo~~ simulated transaction work request; and

determining based on said determined workload, said computer program operation to be performed for said workload.

3. (Previously Presented) The method according to claim 1, wherein said forwarding step further comprises sending a user identification (UID), a user priority and a workload identification (WLID) to the workload driver.

4. (Original) The method according to claim 3, further comprising encapsulating said user identification (UID), said user priority and said workload identification (WLID) in a HTTP header prior to said forwarding step.

5. (Original) The method according to claim 3, further comprising encapsulating said user identification (UID), said priority and said workload identification (WLID) in a URL string prior to said forwarding step.

6. (Previously Presented) The method according to claim 3, wherein said forwarding step further comprises sending said UID, said user priority and said WLID from a user driver to said workload driver.

7. (Previously Presented) The method according to claim 1, wherein said workload driver is a servlet that runs on the e-business application server.

8. (Previously Presented) The method according to claim 7, further comprising instantiating said workload driver by the application server, and wherein said workload driver is a JAVA class.

9. (Currently Amended) A system for application workload simulation on an e-business application server hosting a plurality of e-business application programs, the system comprising:

a user driver for generating ~~placebo~~ simulated transaction work requests having priority indicators, wherein the ~~placebo~~ simulated transaction work request ~~result~~ results in a load being applied to the e-business application server;

a configuration file comprising computer program commands associated with said transaction work requests; and

a workload driver for automatically determining based on configuration information in said configuration file, which of said computer program commands will be executed in the e-business application server based on the priority indicators ~~for said placebo transaction work requests by said user driver~~;

wherein reconfiguring said configuration file varies a processing complexity and duration of at least one of said computer program commands

10. (Original) The system according to claim 9, wherein said workload driver is a servlet instantiated by the application server.

11. (Currently amended) The system according to claim 10, wherein each of the ~~placebo~~ simulated transaction work requests generated by said user driver comprises a user identification (UID), a priority and a workload identification (WLID).

12. (Original) The system according to claim 11, further comprising command classes having varying load processing requirements.

13. (Currently amended) The system according to 9, further comprising a URL string having encapsulated therein, a UID, a priority and a WLID, said URL string for

communicating said ~~placebo~~ simulated transaction work requests from said user driver to said workload driver.

14. (Currently amended) The system according to 9, further comprising a HTTP header having encapsulated therein, a UID, a priority and a WLID, said HTTP header for communicating said ~~placebo~~ simulated transaction work requests from said user driver to said workload driver.

15. (Currently Amended) A machine readable storage having stored thereon, a computer program having a plurality of code sections, said code sections executable by a machine for causing the machine to perform the steps of:

forwarding a ~~placebo~~ simulated transaction work request to the e-business application server hosting a plurality of application programs, wherein the ~~placebo~~ simulated transaction work request includes a priority indicator and results in a load being applied to the e-business application server;

~~applying a load for said placebo transaction work request resulting from said executing step to said e-business application server;~~

receiving said forwarded ~~placebo~~ simulated transaction work request by a workload driver;

said workload driver automatically translating said forwarded ~~placebo~~ simulated transaction work request into [[a]] computer program operation operations to be performed in the e-business application server based on the priority indicator, wherein said automatically translating step is determined by configuration information contained within a configuration file, said configuration file associating computer program commands with transaction work requests, and wherein reconfiguring said configuration file varies a processing complexity and duration of at least one of the computer program commands; and

executing said computer program operation in the e-business application server.

16. (Currently amended) The machine readable storage according to claim 15, wherein said translating step further comprises:

determining a workload based on said forwarded ~~placebo~~ simulated transaction work request; and

determining based on said determined workload, said computer program operation to be performed for said workload.

17. (Previously Presented) The machine readable storage according to claim 16, wherein said forwarding step further comprises sending a user identification (UID), a user priority and a workload identification (WLID) to the workload driver.

18. (Original) The machine readable storage according to claim 17, further comprising encapsulating said user identification (UID), said user priority and said workload identification (WLID) in a HTTP header prior to said forwarding step.

19. (Original) The machine readable storage according to claim 17, further comprising encapsulating said user identification (UID), said priority and said workload identification (WLID) in a URL string prior to said forwarding step.

20. (Previously Presented) The machine readable storage according to claim 17, wherein said forwarding step further comprises sending said UID, said user priority and said WLID from a user driver to said workload driver.

21. (Previously Presented) The machine readable storage according to claim 15, wherein said workload driver is a servlet that runs on the e-business application server.

22. (Previously Presented) The machine readable storage according to claim 21, further comprising instantiating said workload driver by said application server, and wherein said workload driver is a JAVA class.